Kerninduzierte Merkmalsräume

Vorlesung 10, Maschinelles Lernen

Dozenten: Prof. Dr. M. O. Franz, Prof. Dr. O. Dürr

HTWG Konstanz, Fakultät für Informatik

Übersicht

Übersicht

Mathematische Präliminarien: Skalarprodukt

Ein Skalarprodukt oder inneres Produkt auf einem Vektorraum V ist eine symmetrische positiv definite Bilinearform:

1. bilinear:
$$\begin{aligned} \langle x + y \cdot z \rangle &= \langle x \cdot z \rangle + \langle y \cdot z \rangle \\ \langle x \cdot y + z \rangle &= \langle x \cdot y \rangle + \langle x \cdot z \rangle \\ \langle x \cdot \lambda y \rangle &= \lambda \langle x \cdot y \rangle = \langle \lambda x \cdot y \rangle \end{aligned}$$
2. symmetrisch: $\langle x \cdot y \rangle = \langle y \cdot x \rangle$
3. positiv definit: $\langle x \cdot x \rangle \geq 0$, und $\langle x \cdot x \rangle = 0$ genau dann, wenn $x = 0$

Kanonisches Skalarprodukt:

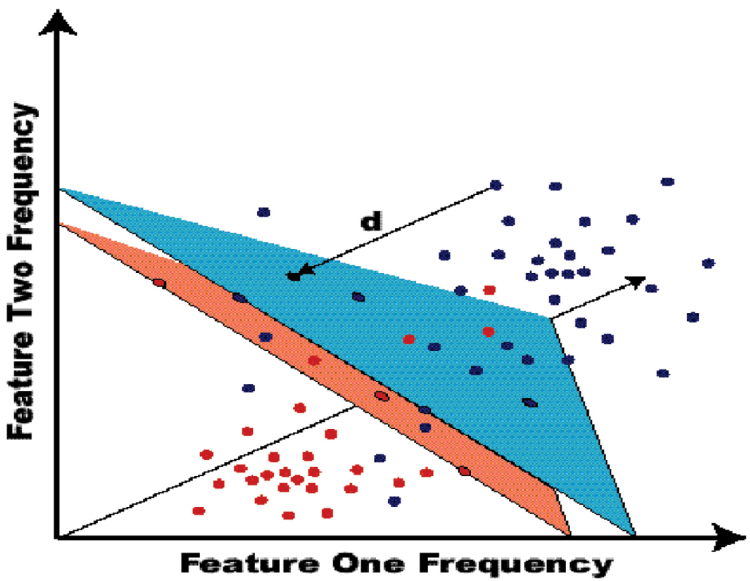
$$(a_1, a_2, \ldots)^\top (b_1, b_2, \ldots) = a_1 b_1 + a_2 b_2 + \ldots$$

Lineare Maschinen

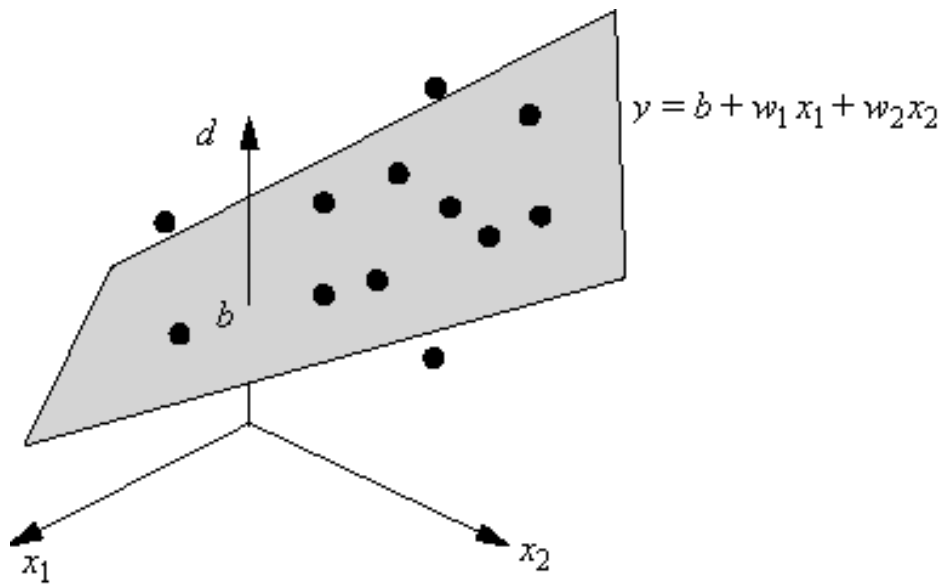
Lineare Lernmaschinen der Form

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \quad \text{mit} \quad \mathbf{x}, \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}$$

definieren eine Trennebene im $\mathbb{R}^d$ (Klassifikation) bzw. einen linearen Zusammenhang zwischen Inputvektoren aus $\mathbb{R}^d$ und skalaren Outputwerten y_i (Regression).



[NeuroSolutions]



[Landry & Winters-Hilt, 2008]

Grenzen des linearen Ansatzes

In vielen Anwendungsfällen reichen lineare Maschinen, die direkt im Eingaberaum operieren, nicht aus:

- Nicht linear trennbare Daten
- Nichtlineare Input-Output-Beziehungen
- Oft ist es sinnvoller, nicht direkt mit den Eingabedaten zu arbeiten, sondern daraus bestimmte **Merkmale** abzuleiten, die leichter zu verarbeiten sind.
- Für bestimmte Probleme (z.B. in der Bioinformatik) ist es nicht sinnvoll, die Datenpunkte als Vektoren darzustellen (z.B. Genomsequenzen).
- Je höher die Dimensionalität des Merkmalsraumes, d.h. je mehr Merkmale aus den Daten abgeleitet werden, desto wahrscheinlicher lassen sich die Daten trennen.

Merkmalsbildung

Ansatz: Statt auf den Eingabedaten wird die lineare Lernmaschine auf die extrahierten Merkmale angewandt. Bei geschickter Wahl der Merkmale wird dadurch das Problem linear trennbar bzw. modellierbar.

Die Merkmalsbildung wird allgemein durch eine Abbildung vom Eingaberaum in den Merkmalsraum (sog. Einbettung)

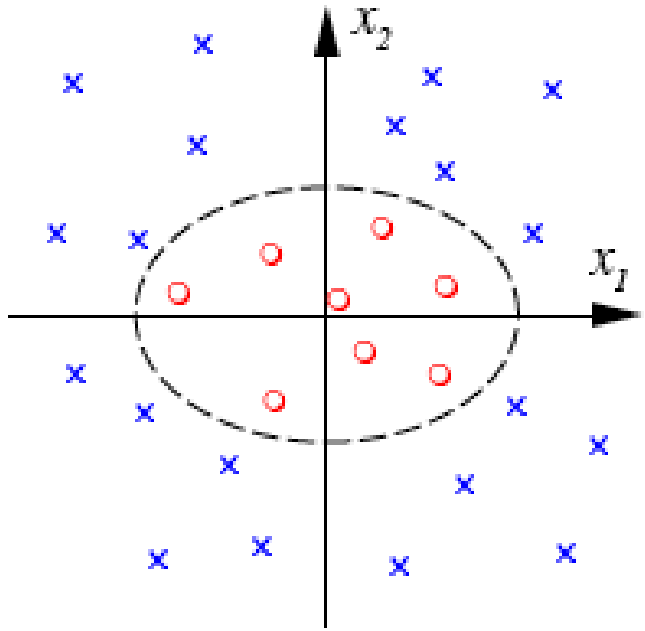
$$\Phi : \mathbb{R}^d \to \mathbb{R}^m \quad \text{mit} \quad \Phi(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \end{pmatrix}$$

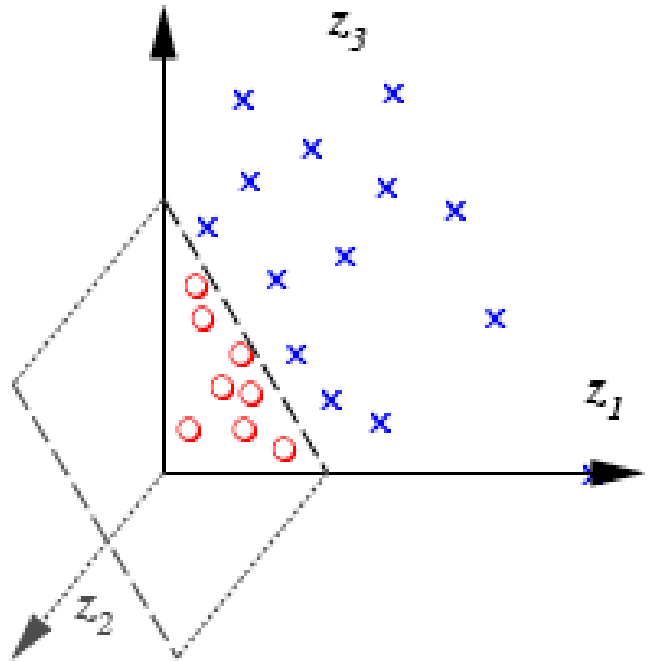
beschrieben.

- Oft ist die Dimensionalität des Merkmalsraumes viel höher als die des Eingaberaumes, manchmal sogar unendlich.
- Jedes Merkmal entspricht einer Dimension im Merkmalsraum, für die jeweils eine eigene Abbildungsfunktion $\phi_i(\mathbf{x})$ definiert werden muß.

Beispiel: Transformation in den Raum der Monome zweiten Grades

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3 \quad \text{d.h.} \quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \quad \text{mit} \quad \Phi(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \phi_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$





Wiederholung: Duale Reprasentation des Gewichtsvektors beim Perzeptron

Der Perzeptron-Algorithmus addiert fehlklassifizierte positive und subtrahiert negative Beispiele. Ist am Anfang $\mathbf{w} = 0$, so ist der resultierende Gewichtsvektor eine Linearkombination der Trainingsdaten:

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i$$

Die positiven Koeffizienten α_i (sog. Einbettungsstärken) sind proportional zu der Anzahl von Fehlklassifikationen, die Beispiel $\mathbf{x}_i$ während des Trainings verursacht hat, und damit eine Art Maß für die Schwierigkeit oder den Informationsgehalt des Datenpunktes.

Für einen festen Datensatz S kann man den Vektor $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots)^\top$ als eine alternative Repräsentation des Gewichtsvektors ansehen, die duale Repräsentation.

Kombination von Merkmalen und linearen Lernmaschinen

Ersetze $\mathbf{x}$ durch $\Phi(\mathbf{x})$ in linearer Lernmaschine (primale Form):

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b, \quad \mathbf{x}, \mathbf{w} \in \mathbb{R}^d \quad \rightarrow \quad f(\mathbf{x}) = \langle \mathbf{w} \cdot \Phi(\mathbf{x}) \rangle + b, \quad \mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^m$$

Aufgrund des Representertheorems sind die Gewichtsvektoren Linearkombinationen der Trainingsdaten $\mathbf{x}_i$, sofern sie Lösung einer großen Gruppe von Optimierungsproblemen sind (s. z.B. Perzeptron):

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i \mathbf{x}_i \quad \text{bzw.} \quad \mathbf{w} = X^\top \boldsymbol{\alpha} \quad \text{mit} \quad X = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \end{pmatrix}, \boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \end{pmatrix}.$$

Die Gewichte α_i (Einbettungsstärken) werden je nach Lernalgorithmus und Aufgabenstellung gewählt.

Duale Form (Anzahl der Gewichte bleibt gleich):

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b \quad \rightarrow \quad f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \langle \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) \rangle + b$$

Skalarprodukt als Ähnlichkeitsmaß

Beispiel Gesichtserkennung: Suche mit Hilfe einer Schablone y_i über Korrelation der Schablone mit den Pixeln x_i eines Suchfensters

$$\sum_{i=1}^{d} y_i x_i$$

d.h. das kanonische Skalarprodukt zweier Muster entspricht ihrer Korrelation.

Damit ergibt sich eine Interpretation der dualen Form einer linearen Maschine

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \langle \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) \rangle + b$$

$\Rightarrow$ Maschine funktioniert über Vergleich der Ähnlichkeit des Inputs zu allen Trainingsbeispielen.

Anwendung auf SVM

Das Representertheorm ist auch für die SVM gültig, d.h. der Gewichtsvektor einer trainierten SVM hat die Form $\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i$. Einsetzen in $\langle \mathbf{w} \cdot \mathbf{w} \rangle$:

$$f(\boldsymbol{\alpha}) = \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle,$$

mit der dualen Entscheidungsfunktion $f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b$ erhalten wir die nur Skalarprodukten abhängige

SVM-Optimierung in dualen Variablen

$$\text{Minimiere } f(\boldsymbol{\alpha}) = \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle, \ \boldsymbol{\alpha} \in \mathbb{R}^{\ell}$$

unter den 2ℓ Nebenbedingungen, $i = 1, \ldots, \ell$

$$\sum_{j=1}^{l} \alpha_j y_j \langle \mathbf{x}_j \cdot \mathbf{x}_i \rangle + b \geq 1 - \xi_i \quad \text{und} \quad \xi_i \geq 0$$

Pegasos für die duale Repräsentation

- Ähnlich wie das Perzeptron ist Pegasos ein **Fehlerkorrekturverfahren**: ein Beispiel $\mathbf{x}_i$ wird nur dann auf den Gewichtsvektor addiert, wenn es die Fehlerbedingung $y_i f(\mathbf{x}_i) < 1$ bzw. $y_i \sum_{i=j}^{l} \alpha_j y_j \langle \mathbf{x}_j \cdot \mathbf{x_i} \rangle$ verletzt.
- Dadurch ergibt sich eine einfache Updateregel: bei jedem fehlklassifizierten Beispiel wird die duale Variable um 1 hochgezählt. Der resultierende Gewichtsvektor muss dann nur noch mit $C \cdot \ \eta_T$ skaliert werden.

Pegasos dual

$$\boldsymbol{\alpha}_{t+1} = \begin{cases} \boldsymbol{\alpha}_t + 1 & \text{falls } y_i f(\mathbf{x}_i) < 1 \\ \boldsymbol{\alpha}_t & \text{sonst} \end{cases}$$

mit $f(\mathbf{x}) = C\eta_t y_i \sum_{i=j}^{l} \alpha_j y_j \langle \mathbf{x}_j \cdot \mathbf{x} \rangle$.

Übersicht

Der Kernel-Trick

Zentrale Beobachtung: In der dualen Form kommen die Trainings- und Testdatenpunkte nur in Form von Skalarprodukten vor, d.h.

$$\langle \mathbf{x}_i \cdot \mathbf{x} \rangle \quad \text{bzw.} \quad \langle \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) \rangle$$

Man kann beweisen, dass dies nicht nur für Kleinste-Quadrate-Regression und Perzeptronen gilt, sondern für eine große Klasse von Optimierungsproblemen (Representer-Theorem, s. Schölkopf & Smola, 2002).

In allen diesen Problemen kann das Skalarprodukt $\langle \Phi(\mathbf{x}) \cdot \Phi(\mathbf{z}) \rangle$ durch eine Kernfunktion

$$k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \quad \text{mit} \quad k_\Phi(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) \rangle$$

ersetzt werden, d.h. Merkmalsbildung und Skalarprodukt werden durch eine Funktionsberechnung ersetzt.

Konstruktion einer Kernfunktion aus der Abbildung in den Merkmalsraum

Die Kernfunktion ist definiert als

$$k_\Phi(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) \rangle,$$

d.h. bei Kenntnis der Abbildung in den Merkmalsraum kann daraus eine Kernfunktion konstruiert werden.

Beispiel: Polynome

$$\begin{aligned}
\langle \Phi(\mathbf{x}) \cdot \Phi(\mathbf{z}) \rangle &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)^\top (z_1^2, \sqrt{2}z_1z_2, z_2^2) \\
&= x_1^2z_1^2 + 2x_1x_2z_1z_2 + x_2^2z_2^2 \\
&= (x_1z_1 + x_2z_2)^2 \\
&= \langle \mathbf{x} \cdot \mathbf{z} \rangle^2
\end{aligned}$$

Allgemein: $\phi_i(\mathbf{x}) = x_{j_1} \cdots x_{j_p}$ für alle Permutationen von $1 \ldots p$ ergibt

$$\langle \Phi(\mathbf{x}) \cdot \Phi(\mathbf{z}) \rangle = \sum_{j_1,\ldots,j_p=1}^{d} x_{j_1} \cdots x_{j_p} z_{j_1} \cdots z_{j_p} = \left(\sum_{j=1}^{d} x_j \cdot z_j \right)^p = \langle \mathbf{x} \cdot \mathbf{z} \rangle^p$$

Mathematischer Einschub: Unendlichdimensionale Vektorräume (1)

Unendlichdimensionale Vektoren können als unendliche mathematische Zahlenfolgen beschrieben werden, z.B.

$$\mathbf{x} = (x_1, x_2, x_3, \ldots, x_i, \ldots) \quad \text{oder} \quad \mathbf{y} = (y_1, y_2, y_3, \ldots, y_i, \ldots).$$

Unendliche Folgen verhalten sich genauso wie Vektoren im $\mathbb{R}^d$, d.h. sie bilden einen Vektorraum V (Folgenraum l):

1. Zwei Folgen können addiert werden und ergeben wiederum eine gültige Folge

$$\mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2, x_3 + y_3, \ldots, x_i + y_i, \ldots) \in V$$

2. Eine Folge kann mit einer Zahl multipliziert werden und ergibt wieder eine gültige Folge

$$a \cdot \mathbf{x} = (ax_1, ax_2, ax_3, \ldots, ax_i, \ldots) \in V$$

Mathematischer Einschub: Unendlichdimensionale Vektorräume (2)

Leider können viele der anderen bekannten Eigenschaften aus dem $\mathbb{R}^d$ nicht auf den Folgenraum übertragen werden, z.B. kann man ohne Einschränkungen der erlaubten Folgen weder die Länge der Vektoren (Norm) noch ein Skalarprodukt definieren. Genausowenig kann man eine Basis für den allgemeinen Folgenraum finden.

Vorgehensweise: Es wird eine Norm für Folgen definiert:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{\infty} |x_i|^p \right)^{1/p}$$

Alle Folgen mit endlicher Norm bilden ebenfalls einen Vektorraum l_p, d.h. Addition und Multiplikation mit einer Zahl führt wieder zu einer Folge mit endlicher Norm. Man sagt, diese Untergruppe von Folgen bildet einen normierten linearen Raum.

Mathematischer Einschub: Unendlichdimensionale Vektorräume (3)

Für den Folgenraum l_2 kann man ein kanonisches Skalarprodukt definieren:

$$\langle \mathbf{x} \cdot \mathbf{y} \rangle = \sum_{i=1}^{\infty} x_i y_i$$

Man kann zeigen, dass dieses Skalarprodukt für jede Kombination von Folgen aus l_2 endlich ist.

Analog zum Endlichdimensionalen mit den beiden Fällen $\langle x \cdot y \rangle = x^\top y$ und $\langle x \cdot y \rangle_S = x^\top S y$ kann man auch hier ein verallgemeinertes Skalarprodukt für eine Folge von nicht negativen Gewichten $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_i, \ldots)$ definieren:

$$\langle \mathbf{x} \cdot \mathbf{y} \rangle_{\boldsymbol{\lambda}} = \sum_{i=1}^{\infty} \lambda_i x_i y_i$$

Lineare Maschinen in unendlichdimensionalen Merkmalsräumen

Voraussetzung für eine funktionierende lineare Maschine: für die Folge $\phi_i(\mathbf{x})$ muß für alle denkbaren Inputwerte $\mathbf{x}$

$$\sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x})^2 < \infty$$

gelten, denn nur unter dieser Voraussetzung ist ein endliches Skalarprodukt für alle Kombinationen von $\mathbf{x}$ und $\mathbf{y}$ garantiert $\Rightarrow$ eine Kernfunktion $k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) \rangle = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$ existiert.

Durch die Kernfunktion läßt sich die Lernmaschine im Dualen durch eine endliche Anzahl von Termen darzustellen:

$$f(\mathbf{x}) = \sum_{i=1}^{\infty} \lambda_i w_i \phi(\mathbf{x}) + b = \sum_{i=1}^{l} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b$$

Übersicht

Bestimmung der Kernfunktion

- Bisher: unter welchen Bedingungen und wie kann man aus einer gegebenen Abbildung in den Merkmalsraum einen Kern konstruieren?
- Umgekehrte Frage: wie kann man überprüfen, ob eine beliebige zweistellige Funktion $k(\mathbf{x}, \mathbf{y})$ eine Kernfunktion ist? $\Rightarrow$ Mercer-Theorem
- Durch das Mercer-Theorem ist es möglich, Kernfunktionen ohne Kenntnis der Einbettung zu konstruieren, z.B. durch die Anwendung bestimmter Kombinationsregeln.

Notwendige Voraussetzung: Symmetrie (wg. Skalarprodukt)

$$k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) \rangle = \langle \Phi(\mathbf{y}) \cdot \Phi(\mathbf{x}) \rangle = k(\mathbf{y}, \mathbf{x})$$

Mercer-Theorem

Mercer-Theorem:

Sei $k(\mathbf{x}, \mathbf{y})$ eine stetige und symmetrische Funktion auf $X \times X$, wobei X eine kompakte Untermenge von $\mathbb{R}^d$ ist.

Falls

$$\int_{X \times X} k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y}) \, d\mathbf{x} d\mathbf{y} \geq 0$$

für alle quadratisch integrierbaren Funktionen $f(\mathbf{x})$ gilt, so kann man $k(\mathbf{x}, \mathbf{y})$ durch folgende Reihenentwicklung darstellen:

$$k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y}) \quad \text{mit} \quad \lambda_i \geq 0.$$

D.h. $k(\mathbf{x}, \mathbf{y})$ ist in diesem Fall eine gültige Kernfunktion mit Einbettung $\Phi = \{\phi_i(x)\}_{i=1}^{\infty}$ und Skalarprodukt $\sum_{i=1}^{\infty} \lambda_i x_i y_i$.

Positiv definitive Kerne

Man kann zeigen, daß aus dem Mercer-Theorem folgende Eigenschaft einer Kernfunktion folgt:

Positiv definite Kerne

$k(\mathbf{x}, \mathbf{y})$ ist genau dann eine gültige Kernfunktion, wenn

- für jede endliche Menge von Trainingsdatenpunkten $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l \in X$
- für jeden Vektor $\mathbf{v} \in \mathbb{R}^l$

$$\mathbf{v}^\top K \mathbf{v} \geq 0 \quad \text{mit} \quad K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

gilt.

D.h. wenn die Gram-Matrix auf jeder denkbaren Trainingsmenge positiv semidefinit ist.

Beispiele für Kernfunktionen

- $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top \mathbf{y})^p$ homogener Polynomkern
- $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + c)^p$ inhomogener Polynomkern
- $k(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \cdot \mathbf{x}^\top \mathbf{y} + \theta)$ Sigmoidkern

  Eine lineare Maschine entspricht einem dreischichtigen Multilayer-Perzeptron (MLP), allerdings erfüllt der Sigmoidkern die Bedingung des Mercer-Theorems nur für bestimmte Werte von κ und θ.
- $k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$ Gaußkern oder RBF-Kern

  Eine lineare Maschine entspricht hier einem RBF-Netzwerk (Radial Basis Function), sehr häufig eingesetzter Standardkern.

Da Kerne einem Skalarprodukt nach der Einbettung im Merkmalsraum entsprechen, kann man sie sich als nichtlineares Ähnlichkeitsmaß vorstellen.

Konstruktion von Kernen aus Kernen

Sind $k_1(\mathbf{x}, \mathbf{y})$, $k_2(\mathbf{x}, \mathbf{y})$ und $k_3(\mathbf{x}, \mathbf{y})$ gültige Kernfunktionen, so sind es auch

- $a \cdot k_1 \quad$ für $\quad a \geq 0 \in \mathbb{R}$
- $k_1 + k_2$
- $k_1 \cdot k_2$
- $f(\mathbf{x}) \cdot f(\mathbf{y}) \quad$ für beliebige Funktionen $f : \mathbb{R}^d \to \mathbb{R}$
- $k_3(\Phi(\mathbf{x}), \Phi(\mathbf{y})) \quad$ für beliebige Abbildungen $\Phi : \mathbb{R}^d \to \mathbb{R}^d$

Eine Reihe weiterer Operationen sind erlaubt, z.B. Tensorprodukte von Kernen, Faltungen, direkte Summen (s. Schölkopf & Smola, 2002).

Kerne für nichtvektorielle Daten

Oft lassen sich die Daten für ein Lernproblem nicht sinnvoll als Vektoren darstellen, z.B. bei der Klassifikation von Texten oder Genomsequenzen. Man kann aber Ähnlichkeitsmaße für solche Objekte definieren, die dann natürlich die Bedingungen des Mercer-Theorems erfüllen müssen, um sinnvoll in einer linearen Lernmaschine eingesetzt werden zu können.

Beispiele:

- Stringkern: Ähnlichkeit zweier Zeichenketten wird über die Häufigkeit und Länge aller gemeinsamen Substrings gemessen, lässt sich rekursiv in linearer Zeit berechnen.
- Ähnlichkeit von Gesten: Produkt der Gelenkwinkel stellt ebenfalls einen Kern da.

Anwendung: Nichtlineare SVM mit Kernfunktion

Mithilfe des Kerneltricks können wir die SVM (und Pegasos) auf einfache Weise in eine nichtlineare Lernmaschine umwandeln:

SVM-Optimierung (nichtlinear)

$$\text{Minimiere } f(\boldsymbol{\alpha}) = \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j),\ \boldsymbol{\alpha} \in \mathbb{R}^{\ell}$$

unter den 2ℓ Nebenbedingungen, $i = 1, \ldots, \ell$

$$\sum_{j=1}^{l} \alpha_j y_j k(\mathbf{x}_j, \mathbf{x}_i) + b \geq 1 - \xi_i \quad \text{und} \quad \xi_i \geq 0$$

Aufgrund der Maximierung der Trennbreite funktioniert die nichtlineare SVM auch in unendlichdimensionalen Merkmalsräumen!